

1 What is claimed is:

2

3 1. A method comprising:

4 storing a firmware module in memory, wherein the firmware module follows a
5 portable executable (PE) format having subdivisions that include an MS-DOS header;
6 and

7 flattening the firmware module by replacing existing content within at least one
8 field within the MS-DOS header of the firmware module with fill data that is more
9 compressible than the existing content.

10

11 2. A method according to claim 1, wherein the operation of flattening the firmware
12 module comprises loading fill data into at least fifty bytes of the MS-DOS header.

13

14 3. A method according to claim 1, wherein the operation of flattening the firmware
15 module comprises loading fill data into an MS-DOS stub field within the MS-DOS
16 header.

17

18 4. A method according to claim 1, wherein the operation of flattening the firmware
19 module comprises ensuring that fill data occupies all fields within the MS-DOS header
20 except for an Ifanew field and an e-magic field.

21

22 5. A method according to claim 1, wherein the PE format also includes an optional
23 file header, the method further comprising:
24 loading fill data into at least one field within the optional file header.

25

26 6. A method according to claim 5, wherein the operation of loading fill data into at
27 least one field within the optional file header comprises:
28 loading fill data into at least one of a SizeOfStackReserve field, a
29 SizeOfStackCommit field, a SizeOfHeapReserve field, a SizeOfHeapCommit field, and
30 a LoaderFlags field.

31

1 7. A method according to claim 1, further comprising:
2 merging at least two sections from an object file into one section in the firmware
3 module.

4

5 8. A method according to claim 7, wherein the operation of merging at least two
6 sections from an object file into one section in the firmware module comprises
7 instructing a linker to merge the at least two sections when generating the
8 firmware module from the object file.

9

10 9. A method according to claim 8, further comprising:
11 causing the linker to change a name of a section specified in the object file to a
12 more compressible name.

13

14 10. A method according to claim 1, wherein the PE format also includes an image
15 page, the method further comprising:

16 storing, in the image page, an alternate file path for a debug file associated with
17 the firmware module, wherein the alternate file path is more compressible than an
18 original file path for the debug file.

19

20 11. A method according to claim 1, wherein the PE format also includes an image
21 page, the method further comprising:

22 instructing a linker to store, in the image page of the firmware module, an
23 alternate file path for a debug file associated with the firmware module, wherein the
24 alternate file path is more compressible than an original file path for the debug file.

25

26

1 12. A program product comprising:
2 a machine accessible medium; and
3 instructions encoded in the machine accessible medium, wherein the
4 instructions, when executed by a processing system, cause the processing system to
5 perform operations comprising:
6 accessing a firmware module within the processing system, wherein the firmware
7 module follows a portable executable (PE) format having subdivisions that include an
8 MS-DOS header; and
9 flattening the firmware module by replacing existing content within at least one
10 field within the MS-DOS header of the firmware module with fill data that is more
11 compressible than the existing content.
12
13 13. A program product according to claim 12, wherein the operation of flattening the
14 firmware module comprises loading fill data into at least fifty bytes of the MS-DOS
15 header.
16
17 14. A program product according to claim 12, wherein the operation of flattening the
18 firmware module comprises loading fill data into an MS-DOS stub field within the MS-
19 DOS header.
20
21 15. A program product according to claim 12, wherein the operation of flattening the
22 firmware module comprises ensuring that fill data occupies all fields within the MS-DOS
23 header except for an Ifanew field and an e-magic field.
24
25 16. A program product according to claim 12, wherein the PE format also includes an
26 optional file header, the program product further comprising:
27 instructions which, when executed by the processing system, cause the
28 processing system to load fill data into at least one field within the optional file header.
29

1 17. A program product according to claim 16, wherein the operation of loading fill
2 data into at least one field within the optional file header comprises:
3 loading fill data into at least one of a SizeOfStackReserve field, a
4 SizeOfStackCommit field, a SizeOfHeapReserve field, a SizeOfHeapCommit field, and
5 a LoaderFlags field.

6
7

1 18. A processing system with resources for flattening a firmware module, the
2 processing system comprising:
3 a processor;
4 memory communicatively coupled to the processor; and
5 instructions stored in the memory, wherein the instructions, when executed by
6 the processor, cause the processing system to perform operations comprising:
7 accessing a firmware module within the processing system, wherein the firmware
8 module follows a portable executable (PE) format having subdivisions that include an
9 MS-DOS header; and
10 flattening the firmware module by replacing existing content within at least one
11 field within the MS-DOS header of the firmware module with fill data that is more
12 compressible than the existing content.
13
14 19. A processing system according to claim 18, wherein the operation of flattening
15 the firmware module comprises loading fill data into at least fifty bytes of the MS-DOS
16 header.
17
18 20. A processing system according to claim 18, wherein the operation of flattening
19 the firmware module comprises loading fill data into an MS-DOS stub field within the
20 MS-DOS header.
21
22 21. A processing system according to claim 18, wherein the operation of flattening
23 the firmware module comprises ensuring that fill data occupies all fields within the MS-
24 DOS header except for an Ifanew field and an e-magic field.
25
26 22. A processing system according to claim 18, wherein the PE format also includes
27 an optional file header, the processing system further comprising:
28 instructions which, when executed by the processor, cause the processing
29 system to load fill data into at least one field within the optional file header.
30

1 23. A processing system according to claim 22, wherein the operation of loading fill
2 data into at least one field within the optional file header comprises:
3 loading fill data into at least one of a SizeOfStackReserve field, a
4 SizeOfStackCommit field, a SizeOfHeapReserve field, a SizeOfHeapCommit field, and
5 a LoaderFlags field.

6
7

1 24. An apparatus comprising:
2 a machine accessible medium; and
3 a firmware module encoded in the machine accessible medium, the firmware
4 module having a portable executable (PE) format with subdivisions that include an MS-
5 DOS header, wherein the firmware module was produced by operations comprising:
6 flattening the firmware module by replacing existing content within at least one
7 field within the MS-DOS header of the firmware module with fill data that is more
8 compressible than the existing content.
9
10 25. An apparatus according to claim 24, further comprising:
11 a processor communicatively coupled to the machine accessible medium;
12 memory communicatively coupled to the processor; and
13 instructions stored in the memory, wherein the instructions, when executed by
14 the processor, cause the processing system to perform operations comprising:
15 retrieving the firmware module from the machine accessible medium; and
16 executing the firmware module within a boot environment.
17
18 26. An apparatus according to claim 24, wherein:
19 the machine accessible medium comprises a non-volatile storage device; and
20 the apparatus further comprises an interface in communication with the non-
21 volatile storage device, the interface operable to provide communication between the
22 non-volatile storage device and a processor of a data processing system.
23
24 27. An apparatus according to claim 26, wherein the apparatus comprises an
25 adapter card for a processing system.
26